EVALUATING TERMINAL HEADS OF LENGTH K

ABSTRACT. This paper presents an alternative algorithm for finding the terminal heads of length k of a given string in a given context-free grammar for given k.

The algorithm given here is an alternative to that provided by [1]. It takes a very different approach to the problem of evaluating the terminal heads of length k of a given string in a given context-free grammar for given k.

Algorithm

Let the given string whose terminal heads of length k we wish to evaluate be $\mathbf{x_1x_2...x_n}$. A string is said to **vanish** if it derives the null string

Let **H** and **S** be sets of strings initially empty. At the termination of the algorithm H will contain the required heads of length K of $x_1x_2...x_n$, while S will contain the set of terminal strings of length < k which are derivable from $x_1x_2...x_n$.

Let **L** be an auxiliary ordered list of strings which initially consists just of the head $x_1x_2...x_n$ up to the kth symbol that does not vanish, or all of $x_1x_2...x_n$ if it contains less than k symbols that do not vanish.

for $1 \le j \le n$ and while L is not empty, do:

> for i = 1, 2, ... until the last member of L is the i_{th} one, and no new members are added to L by the following procedure, do:
>
> > add to the end of L the result of applying all possible productions to the jth symbol in the ith member of L, omitting strings that are already in L, and truncating all members added which have k or more symbols that do not vanish, by deleting the part of the string following the kth symbol that does not vanish
>
> next i
>
> Remove from L all strings whose jth symbol is a nonterminal.
> Remove from L all strings which possess k-heads consisting entirely of terminals, and add the k-heads involved to the set H.
> Remove from the list all strings of length < k which consist entirely of Terminal and add these to the set S

next j

EXAMPLE

Given the grammar (where λ is the null string)

X → Y | x | λ
Y → Z | y | λ
Z → X | z | λ
U → u

to find the set of terminal 2-heads of XYZU

		item no.	list L
j=1	i=1	1	XYZU
		2	YYZU
		3	xYZU
		4	YZU
	i=2	5	ZYZU
		6	yYZU
		(we don't add YZU here as it has already occurred.)	
	i=3		
	i=4	7	ZZU
		8	yZU
		9	ZU
	i=5	10	zYZU
	i=6		
	i=7	11	XZU
		12	zZU
	i=8		
	i=9	13	XU
		14	zU
		15	U
	i=10		
	i=11	16	xZU
	i=12		
	i=13	17	YU
		18	xU
	i=14		
	i=15	19	u
	i=16		
	i=17	20	yU
	i=18		
	i=19		
	i=20		

We now remove all strings from L with nonterminals in their jth (i.e. first) position, which leaves us with the items 1-8 below, and remove item 19 and set S = {u}

j=2	i=1	1	xYZU	
		2	yYZU	
		3	yZU	
		4	zYZU	
		5	zZU	
		6	zU	
		7	xZU	
		8	xU	
	i =1	9	xZZU	
		10	xy	(truncated from xyZU)
	i=2	11	yZZU	
		12	yy	
	i=3	13	yXU	
		14	yU	
	i=4	15	zZZU	
		16	zy	
	i=5	17	zXU	
		18	zz	
	i=6	19	zu	
	i=7	20	xXU	
	i=8	21	xu	
	i=9	22	xXZU	
		23	xz	
	i=11	24	yXZU	
		25	yz	
	i=12			
	i=13	26	yx	
	i=14	27	yu	
	i=15	28	zXZU	
	i=16			
	i=17	29	zYU	
		30	zx	
	i=20	31	xYU	
		32	xx	

Removing all strings with nonterminals in the 2nd position, provides the set H required:

xy
yy
zy
zz

zu
xu
xz
yz
yx
yu
zx
xx

REFERENCE

1. Alfred V. Aho and Jeffrey D. Ullman. The Theory of Parsing , Translation, and Compiling, vol. 1. Prentice-Hall, 1972.